# BombAppetit

SIRS-23/24

# BombAppetit

BombAppetit is a user-friendly web app that makes restaurant reservations easy. It's designed to simplify the process with a straightforward interface. The app also offers a voucher service, letting users use them for great booking discounts. BombAppetit is all about making dining hassle-free and connecting users with enjoyable culinary experiences.

# Secure document format

The document is secured in this steps:

1- To ensure authenticity:

1.1- We added a nonce inside the property “auth” in a new property called “counter”, to ensure freshness

1.2- Create and add a DS :

- The DS is calculated using RSA to encrypt and SHA256 to calculate the hash
- The DS is then added in the property “auth” in a new property called “DS”

# Secure document format

2- To ensure confidentiality:

2.1- We generate a  symmetric key (session key) and a IV

2.2- Remove the the “vouchers“ property from the json

2.3- The vouchers need to be confidential, so to do that we encrypt them using a symmetric key (session key)

- And we add the base64 encode of them to the json with a property name “vouchers”

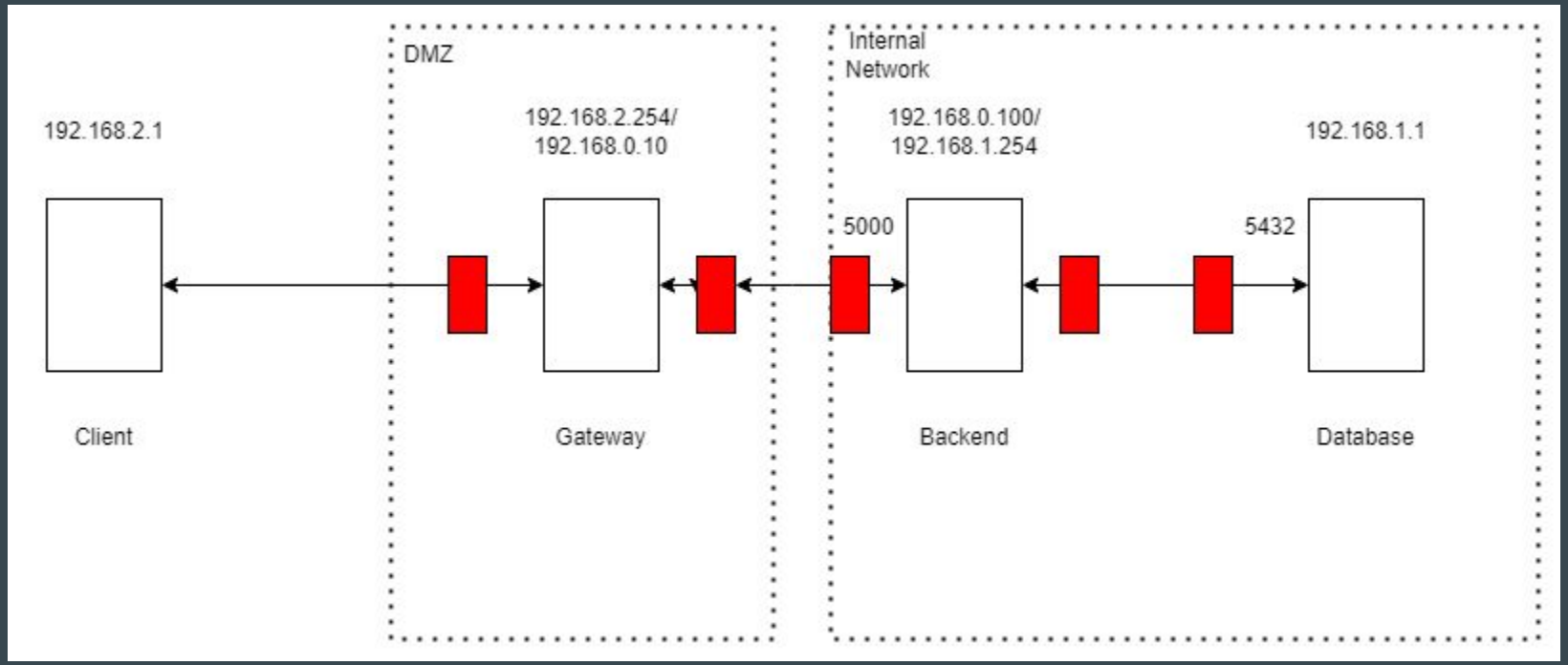
2.4- This session key also needs to be passed to the client, encrypted using the client public key

- And we add the base64 encode of them to the json  in the property “auth”  with a property name “encSymKey”

2.5- This IV  also needs to be passed to the client, encrypted using the client public key

- And we add the base64 encode of them to the json  in the property “auth”  with a property name “encIV”

# Infrastructure



# Infrastructure

The gateway is in a DMZ zone. Redirects the traffic from the client to the backend and vice-versa. The gateway only redirects traffic from the backend to the client if it is a response to a request from the client, and only redirects traffic to the backend on port 5000. Rejects all other traffic.

The backend is in the internal network. Accepts traffic from the DB from port 5432 and tcp traffic to port 5000. The traffic from the DB is only accepted if the connection is not new. The backend sends traffic to the DB to port 5432, and only sends traffic to other ips if the connection is already established. Rejects all other traffic.

The database is also in the internal network. Only accepts traffic from the server to port 5432. And only sends traffic to the server if the connection is already established. Rejects all other traffic.

# Secure channels and key distribution

For the communication between the client and the server, we use https.

https ensures that the communication is confidential and that the client is talking to the correct server. The client has the server certificate and the backend has the client certificate. The client also has the CA certificate to verify the server certificate.

The Backend and the Database communicate using TLS, to ensure the confidentiality of the communication.

Client has his own private key, public key and the server public key. The server has his own public and private key, and the client public key. We assume the client already exchanged keys with the server, so the server has the client public key.

# Security Challenge

- Introduce Reviews
  - should be non-repudiable
  - other users must be able to verify the authenticity of each review

# Security Challenge

- Allow users to directly transfer vouchers to other users

The server changes the client associated with the voucher

- but only if the voucher is owned by the client that is sending the request and
- if the voucher was not already used.

# Security Challenge

The reviewers are sent by the server, encrypted using the session key (symmetric key) and then encoded in base64, like the vouchers.

Sending the vouchers to another client only requires the server to change the client associated with the voucher in the database.

# Demonstration